Appendix B: Compliance Checklists

DII compliance is the cornerstone to ensure seamless segment integration and proper system operation. An objective technique for measuring DII compliance is required. Such a system has the following advantages:

- It allows quantitative statements to be made about whether or not a segment or COE-based system is compliant. It also provides an objective measure of the *degree* to which a segment or system is DII-compliant.

- It serves to identify areas in which segments need to improve to achieve compliance. A side benefit is the identification of potential areas of interoperability problems because of identifying areas where the system overlaps COE functionality.

- It provides a meaningful way to quantitatively compare segments in addition to traditional measures such as functional coverage. One important dimension to this technique is that it directly incorporates interoperability comparisons between segments being evaluated.

- It aids in developing a migration strategy for legacy systems. Program managers can use the checklist to determine where they stand with regards to compliance, and then use the identified areas of non-compliance to create a strategy and schedule for achieving the target level of compliance.

This appendix addresses primarily Category 1 (Runtime Environment) compliance, although there are also certain Category 2 (Style Guide) compliance items listed. The *DII Style Guide* addresses Category 2 compliance and also has a checklist organized to compliment the checklist given here. Refer to that document for specific Category 2 requirements and for how they map to the equivalent Category 1 levels presented here.

Chapter 2 defines Category 1 compliance as eight levels of progressively deeper integration, because compliance cannot be an all or nothing proposition for legacy systems. The levels progress from a state of "peaceful coexistence" to "federation of systems" to true integration. Chapter 2 also shows how the levels of compliance map to levels of interoperability, and that interoperability increases as the level of DII compliance increases. As noted in Chapter 2, segments shall achieve Level 7 compliance before being accepted or advertised as an approved DISA product. At DISA's discretion, segments that are Level 5 or Level 6 compliant may be accepted as prototypes and fielded at selected sites for evaluation purposes. Developers must achieve Level 4 compliance before DISA will consider evaluating a prototype for eventual migration into the COE or COE-based DISA systems.

This appendix contains a series of questions, in a checklist format, which are organized by compliance level. The philosophy behind this design for the compliance checklists is to

begin with an agreement on a set of standards[1], ensure non-interference when installed on the same LAN, then non-interference when installed on the same workstation, and finally to interoperability through sharing the same software and data. With this strategy, it is possible to define a minimally acceptable level of compliance that balances system risk against cost to achieve full compliance.

Segments shall be evaluated against these checklists to determine the degree of compliance. There are several things to note in order to properly apply the compliance checklists in this appendix.

1. System components are not required to be in segment format until Level 4. Therefore, the compliance checklists use the generic term "application" to refer to any system component being evaluated in levels 1-3. Beginning with Level 4, they are properly referred to as segments because compliance at Level 4 requires segmentation.

2. Each item in the checklists shall be answered as *True*, *False*, or *Not Applicable* as appropriate. Not all questions are necessarily applicable to all segments, and hence should be marked as "N/A." For example, questions that deal with database issues are not applicable to applications/segments that do not use a database. Similarly, questions that deal with Motif are not applicable to applications/segments that run on NT platforms, or which do not provide a GUI.

3. The Category 1 compliance level assigned to the application/segment is the highest numbered level for which there are no "False" replies. As explained in Chapter 2, it is not permissible to describe an application/segment in a manner such as "80% Level 6" compliant.

4. Note that each question is prefaced by the level it applies to. Thus, question 6-3 refers to the third compliance question for Level 6.

5. Each question in the checklists is independent of the operating environment (e.g., Unix versus NT) unless noted as environment-specific. Environment-specific questions are noted by including "(Unix)" or "(NT)" as appropriate at the beginning of the question.

6. Each successive level is inclusive. For example, Level 6 compliance means that the segment has achieved compliance for levels 1-5.

7. In the checklists there are references made to approvals by the Chief Engineer. Unless otherwise qualified, this means the DII COE Chief Engineer for COE-component segments, or the cognizant program Chief Engineer for mission-application segments. The principle being applied is that any modification that affects interoperability or the COE requires approval by the DII COE Chief Engineer. All other approvals are the domain of the program Chief Engineer because they are limited in scope to the system being built.

[1] The *JTA* contains a standards profile that compliant systems must adhere to. However, to allow for migration of legacy systems, full compliance is not stipulated in Level 1 but is spread judiciously across all 8 levels.

8. In the checklists there are references to the SSA. Unless otherwise qualified, this means the DII COE SSA for COE-component segments, or the cognizant program SSA for mission-application segments.

9. In each case, it is the intent that "sanity is to prevail." Applications/segments are expected to comply with the requirements specified, but the DII COE Chief Engineer may grant unusual exceptions on a case-by-case basis.

10. The COE is presently available for NT only on Intel-based platforms (e.g., 80x86, Pentium). The NT questions in the checklist are applicable only to that hardware environment and will be upgraded as required if NT support is made available for non-Intel platforms. The concept conveyed by the checklist items are applicable regardless of the hardware platform or operating system, but for clarity are often worded in such a way as to make the statement operating-environment-specific.

In general, applications should be able to reach Level 5 compliance without requiring source code changes. This is true only if good programming practices are followed. For example, use of hardcoded pathnames is not a good programming practice. It would thus likely require source code changes to achieve Level 5 compliance because of the requirement that the segment use relative filenames for files within the segment (see the *Runtime Environment* series of questions below under Level 5 compliance).

The checklists presented in this appendix are organized in a very deliberate way and with a deliberate objective in mind of evaluating DII compliance. There are several points about how the checklists are organized.

- The checklists are organized in such a way as to evaluate individual segments. However, as Chapter 2 describes, the compliance of individual segments can be combined into a composite compliance level. Many of the items in the appendix are applicable to only specific types of segments (e.g., COE-component segments, account group segments), and should be noted as "not applicable" for many systems.

- The organization presented here is not the only possible, nor the only useful, way of organizing a compliance checklist. It could be organized by segment type, but such an approach leads to duplication of checklist items among the various segment types. It could also be organized by how the segment will be used in the resulting system (e.g., user presentation, data server, application server), but this suffers the same problem as organization by segment type and moreover, the compliance criteria for a segment should be independent of how the segment will ultimately be used in the target system.

- The organization presented here does not directly address the amount of work required to reach any specific level of compliance. The reason it does not attempt to do so is because the effort required to reach compliance is heavily dependent upon the system under evaluation, and the areas in which the system is weak with respect to compliance. For example, a Unix system that is not based on Motif will likely require a

substantial amount of effort to achieve Level 7 or 8 compliance even though there are only a few checklist questions related to Motif. On the other hand, there are several specific questions related to how the system is packaged into segments, but this generally requires little or no coding changes at all to achieve compliance.

- The organization presented here is conducive to pinpointing problem areas both from a compliance perspective, and as a side effect, from an interoperability perspective. Part of the effort in achieving compliance is to ensure conformance to standards, and to ensure that there is no duplication of COE services. This directly leads to pinpointing potential interoperability problem areas.

- The COE specifically avoids weighting compliance questions. This has the obvious disadvantage of assigning equal importance to each question within a level, but the reverse problem is that assigning weights cannot be done outside the context of the objectives of the end system. For example, if the objective is to migrate a legacy system rapidly so that it can be deployed quickly, that leads to a different set of weights than if the objective is to achieve full interoperability.

Program managers may wish to organize the compliance checklists presented here in a manner more conducive to meeting programmatic objectives. However, the compliance levels shall not be modified, nor the applicable questions at each level.

B-1. Standards Compliance (Level 1)

Standards Compliance				
T	**F**	**N/A**	**1-1**	(NT) Hardware components are Windows NT-compliant as defined by the Microsoft document *Microsoft Windows NT Hardware Compatibility List #4094*.
Operating System				
T	**F**	**N/A**	**1-2**	The operating system is FIPS 151-2 POSIX.1-compliant.
T	**F**	**N/A**	**1-3**	Unless approved by the DII COE Chief Engineer, the operating system supports the `System` API for FIPS 119 (Ada95).
T	**F**	**N/A**	**1-4**	The operating system is configured to support DCE.
T	**F**	**N/A**	**1-5**	The operating system is configured to support TCP/IP protocols.
T	**F**	**N/A**	**1-6**	The operating system is configured to support UDP protocols.
T	**F**	**N/A**	**1-7**	The operating system is configured to support SLIP and PPP.
T	**F**	**N/A**	**1-8**	(Unix) The operating system is based on System V Unix.
T	**F**	**N/A**	**1-9**	(NT) The operating system is the same version as provided by the COE, or higher (see Appendix A).
Network Services				
T	**F**	**N/A**	**1-10**	The application can execute in an environment that includes DII COE-provided DCE services.
T	**F**	**N/A**	**1-11**	The application uses only those TCP/IP interfaces provided by the native operating system.
T	**F**	**N/A**	**1-12**	The application uses only those UDP or point-to-point interfaces provided by the native operating system.
T	**F**	**N/A**	**1-13**	The application uses only those SLIP or PPP interfaces provided by the native operating system.

GUI Environment				
T	**F**	**N/A**	**1-14**	The application complies with the style of the native GUI. (See GUI compliance requirements in the *DII Style Guide*.)
Database Services				
T	**F**	**N/A**	**1-15**	If an RDBMS is used, it supports FIPS-127-2 SQL queries.

B-2. Network Compliance (Level 2)

Security				
T	**F**	**N/A**	**2-1**	The application is able to operate correctly with the operating system security modules enabled (BSM for Solaris, C2 enabled for HP, etc.).
Operating System				
T	**F**	**N/A**	**2-2**	The operating system supports NFS servers and clients.
T	**F**	**N/A**	**2-3**	(Unix) The operating system can be configured to support DNS/NIS/NIS+. (Note: The requirement is that the operating system be capable of supporting centralized management of key resources such as hostnames, user accounts, etc. NIS+ is not a specific requirement because not all vendors support it.)
T	**F**	**N/A**	**2-4**	(NT) NT is configured to use the NTFS file system for files stored on hard disks. (Note: NT uses the FAT file system for floppy diskettes. Such usage is generally transparent to applications. However, NTFS is required on the hard disk for security reasons.)
Network Services				
T	**F**	**N/A**	**2-5**	The operating system supports sockets, including Berkeley sockets.
T	**F**	**N/A**	**2-6**	The application is able to operate properly in an environment where other applications are performing UDP broadcasts.
T	**F**	**N/A**	**2-7**	The application does not require any particular hostname conventions nor does it need reserved IP addresses.
T	**F**	**N/A**	**2-8**	The ability of the application to execute correctly is independent of the type of LAN (e.g., Class B or Class C) connected to the platform.
T	**F**	**N/A**	**2-9**	(Unix) The application can operate in a DNS/NIS/NIS+ environment. (Note: The requirement is that the application be able to operate correctly when the features supported by the operating system for centralized management of key resources are enabled.)
T	**F**	**N/A**	**2-10**	(NT) If the target system is configured to use Microsoft domains and workgroups, the application can operate correctly in such an environment.
T	**F**	**N/A**	**2-11**	(NT) The application uses native PC byte order for data internal to the PC, but uses network byte order for data external to the PC.
T	**F**	**N/A**	**2-12**	(NT) The application uses native PC byte order to access `$DATA_DIR/local` and `$DATA_DIR/PCglobal` PC data.
T	**F**	**N/A**	**2-13**	(NT) The application uses network byte order to access `$DATA_DIR/global` data.

GUI Environment				
T	**F**	**N/A**	**2-14**	(Unix) If the application resides on a machine with an X server, the X server is compatible with the version supplied by the COE (see Appendix A).
Database Services				
T	**F**	**N/A**	**2-15**	Database updates operate correctly with DBMS security audits enabled.
T	**F**	**N/A**	**2-16**	The database is recoverable to a consistent state in the event of DBMS server, network, or client application failure. This includes both hardware and software failures.
T	**F**	**N/A**	**2-17**	Database transactions implement strict two-phase locking.

B-3. Workstation Compliance (Level 3)

Operating System				
T	**F**	**N/A**	**3-1**	If extensions to the operating system as configured for the COE are required, all such extensions have been identified and document. This includes the configuration of all operating system resources including the amount of shared memory required, the number of semaphores, the message queue size, etc.
T	**F**	**N/A**	**3-2**	The operating system configuration required by the application does not decrease or con flict with any system resources as already configured for the COE. The application may increase system resource configurations, but not decrease them.
T	**F**	**N/A**	**3-3**	The application does not use hardcoded port assignments (e.g., from `/etc/services`) and is not sensitive to specific ports other than well-known port assignments (e.g., `ftp`, `listen`).
Network Services				
T	**F**	**N/A**	**3-4**	The application can operate in an environment configured to only use anonymous ftp.
GUI Environment				
T	**F**	**N/A**	**3-5**	(Unix) The application does not make direct calls to X libraries that conflict with applications that use Motif libraries to access lower-level X functions. For example, the application does not use lower-level X library functions to establish window border style or colors that either conflict or override settings established by Motif.
T	**F**	**N/A**	**3-6**	(Unix) The application does not alter any files in the vendor-supplied X or Motif directories (e.g., modify `rgb.txt` or `Xdefaults`) unless authorized by the DII COE Chief Engineer. Approval by the DII COE Chief Engineer is required because of the potential impact of the presegmented application on other segments running on the same workstation.
T	**F**	**N/A**	**3-7**	(Unix) The application can use the same X server version and `xdm` version that is supplied by the COE (see Appendix A).
T	**F**	**N/A**	**3-8**	(Unix) The application uses either the same version of Motif as provided by the COE (see Appendix A), or does a static link to Motif libraries so that it does not conflict with other COE-based segments. The specific Motif version required is identified in the *DII Style Guide*.
T	**F**	**N/A**	**3-9**	(NT) The application uses the same version of NT as supplied by the COE (see Appendix A).
T	**F**	**N/A**	**3-10**	(NT) Unless a COTS application, the application uses only Win32 APIs to access Windows routines.

Database Services				
T	**F**	**N/A**	**3-11**	The application does not modify the user's DBMS environment as established by the DBMS COE-component segment.
DCE Services				
T	**F**	**N/A**	**3-12**	If using RPCs, the application is compatible with the RPC mechanisms supported by the DCE version supplied by the COE.
COTS Products				
T	**F**	**N/A**	**3-13**	The software is capable of running in an environment that includes DII COE approved COTS products as specified in the *DII COE Baseline Document* for the COE version being used.
T	**F**	**N/A**	**3-14**	Configuration changes made to COTS products, if any, do not render inoperable any features available to COE-based segments or users that are already using the COTS product.
T	**F**	**N/A**	**3-15**	The application does not require any source code modifications to COTS products, except as authorized by the DII COE Chief Engineer. (Some commercial products, such as `xdm`, may require modification for security reasons. `xdm` has already been modified by the DII COE Chief Engineer to address security concerns.)
T	**F**	**N/A**	**3-16**	(NT) If the application is a COTS product that uses 16-bit APIs, there is no 32-bit alternative.
Runtime Environment				
T	**F**	**N/A**	**3-17**	The application does not alter any files outside its own directory in such a way that it conflicts with any other COE-based segment.
T	**F**	**N/A**	**3-18**	The application can operate on a COE-configured workstation without altering the location or version of any system software (Unix, X Windows, Motif, NT, etc.).
Miscellaneous				
T	**F**	**N/A**	**3-19**	(NT) The application supports VGA and SVGA resolutions.
T	**F**	**N/A**	**3-20**	(NT) The application supports 16x16, 32x32, and 64x64 icons.

B-4. Bootstrap Compliance (Level 4)

Security				
T	**F**	**N/A**	**4-1**	If an aggregate segment, the security level of the parent dominates the security level of the children.
T	**F**	**N/A**	**4-2**	Documentation is submitted with the segment that clearly identifies releasability restrictions.
Standards Compliance				
T	**F**	**N/A**	**4-3**	All software and data are packaged in segment format.
T	**F**	**N/A**	**4-4**	The segment successfully passes `VerifySeg` with no errors.
T	**F**	**N/A**	**4-5**	The segment uses the same bootstrap COE as provided by the COE, or all extensions required are documented and handled by the segment in such a way that it does not interfere with other segments. For example, community files are not destructively overwritten by the segment because other segments may also need to made alterations to the community file during their own installation.
T	**F**	**N/A**	**4-6**	The segment can be installed and removed completely through the COE installation tools. If the segment is a "permanent" segment (see Chapter 5) and is not a candidate for removal, the segment has been tested to ensure that upgrades successfully preserve data files that must be retained during upgrades.
T	**F**	**N/A**	**4-7**	(NT) Unless a COTS segment, the segment does not modify the root-level `AUTOEXEC.BAT` or `CONFIG.SYS` files.
T	**F**	**N/A**	**4-8**	(NT) Unless a COTS segment, the segment does not modify any Windows `INI` files.
Database Services				
T	**F**	**N/A**	**4-9**	Database owners do not use system storage areas during database creation.
T	**F**	**N/A**	**4-10**	The segment does not modify the core database storage areas, create objects in system storage areas, or create objects in public storage areas (e.g., create rollback table space).
COTS Products				
T	**F**	**N/A**	**4-11**	The segment uses the same COTS configurations as those specified by the applicable *DII COE Baseline Document* for any COTS product it uses that may also reside on the workstation.

Runtime Environment				
T	**F**	**N/A**	**4-12**	Runtime extensions to the COE required by the segment have been identified and documented.
T	**F**	**N/A**	**4-13**	The segment uses the same runtime environment configuration as provided by the COE with extensions, if any, made through environment extension files and segment descriptors.
T	**F**	**N/A**	**4-14**	The segment uses the same versions, configurations, patches, and file locations as provided by the COE for all components of the bootstrap COE.
T	**F**	**N/A**	**4-15**	The segment uses the DII COE directory layout or a migration plan to achieve proper directory layout has been prepared.
T	**F**	**N/A**	**4-16**	(NT) The segment is able to handle Unicode filenames.

B-5. Minimal DII Compliance (Level 5)

Security				
T	**F**	**N/A**	**5-1**	For COE-component segments, prior approval has been granted by the DII COE Chief Engineer to provide a command-line mode or feature. The `$CMDLINE` keyword is used in the `Direct` segment descriptor to indicate command-line access is provided.
T	**F**	**N/A**	**5-2**	For mission-application segments, p rior approval has been granted by the Chief Engineer to provide a command-line mode or feature. The `$CMDLINE` keyword is used in the `Direct` segment descriptor to indicate command-line access is provided.
T	**F**	**N/A**	**5-3**	For all segments, whether COE-component segments or mission-application segments, prior approval has been granted by the DII COE Chief Engineer to provide a command-line mode or feature that provides "superuser" access. The `$CMDLINE` and `$SUPERUSER` keywords are used in the `Direct` segment descriptor to indicate superuser access.
T	**F**	**N/A**	**5-4**	The segment does not provide a "back door" access to a command-line prompt. If a command-line mode is available, it is through a known, documented approach for all authorized users and not through some hidden, undocumented approach.
T	**F**	**N/A**	**5-5**	If privileged user permissions are required during segment installation or removal (e.g., use of the `$ROOT` keyword), prior approval has been granted by the Chief Engineer.
T	**F**	**N/A**	**5-6**	(Unix) The segment does not alter the COE established `umask` setting.
Standards Compliance				
T	**F**	**N/A**	**5-7**	The segment uses the same kernel COE as provided by the COE and documented in the applicable *DII COE Baseline Document* for the COE version being used.
T	**F**	**N/A**	**5-8**	All directory and file names contain only p rintable, non-blank, standard ASCII characters.
T	**F**	**N/A**	**5-9**	The segment does not create user login accounts. (This does not apply to the account group segments that are part of the kernel COE, but it *does* apply to all other account group segments.)
T	**F**	**N/A**	**5-10**	The segment can operate in an environment where user accounts are created and deleted at any time by the site administrator responsible for managing user accounts. The segment accounts for this and creates and initializes operator preferences the first time the segment is activated after a new account is created.
T	**F**	**N/A**	**5-11**	The segment loads correctly into the directory assigned by the COE installation tools. It does not require being loaded in any specific directory, or the Chief Engineer has granted a waiver. (This requirement does *not* apply to COTS segments.)
T	**F**	**N/A**	**5-12**	The segment conforms to the COE version numbering scheme.

T	**F**	**N/A**	**5-13**	The segment does not move directories or files from the segment's home directory into other directories unless approved by the DII COE Chief Engineer. (This requirement is stipulated to avoid circumventing the intent of the *I&RTS* by loading the segment as directed by the installer, and then moving the segment to some other location during `PostInstall`.) This requirement does *not* apply to COTS segments, nor does it apply to data that is being moved to the proper `$DATA_DIR/global` or `$DATA_DIR/local` directory.
T	**F**	**N/A**	**5-14**	(NT) The segment creates all its subkeys underneath *`SegType`*`\`*`SegDirName`* where *`SegType`* is `Account Groups`, `COE`, `COTS`, `Patches`, `Data`, or `Software`, and *`SegDirName`* is the segment's directory name.
T	**F**	**N/A**	**5-15**	(NT) Unless a COTS segment, the segment does not create any root keys.
T	**F**	**N/A**	**5-16**	(NT) All segment subkeys are named with the segment prefix.
T	**F**	**N/A**	**5-17**	(NT) The segment supports UNC filenames.
Operating System				
T	**F**	**N/A**	**5-18**	The segment does not rename well defined ports (e.g., `ftp`, `listen`), or declare new port names which have the same port number as well-defined ports in the Unix `/etc/services` file, or the NT equivalent of this file.
T	**F**	**N/A**	**5-19**	If ports are required, the quantity has been identified and documented in the `COEServices` segment descriptor.
GUI Environment				
T	**F**	**N/A**	**5-20**	The segment is fully compliant with the style of the native GUI (see compliance requirements in the *DII Style Guide*).
T	**F**	**N/A**	**5-21**	The segment uses the window manager provided by the COE (`dtwm`[2] for Unix, Windows NT for NT platforms).
T	**F**	**N/A**	**5-22**	(Unix) The segment is compatible with the `XFONTSDIR`, `XAPPLRESDIR`, and `XENVIRONMENT` settings established by the COE.

[2] With the present *I&RTS* release, a commercial CDE product provides the desktop. Thus, `dtwm` replaces `mwm` from the previous *I&RTS*. There should not be any impact to any segment that presently works under `mwm`.

Database Services				
T	**F**	**N/A**	**5-23**	Application segments are separate from their corresponding database segment.
T	**F**	**N/A**	**5-24**	Application segments that access databases operate correctly from any COE-compliant workstation and are not required to be installed on a database server.
T	**F**	**N/A**	**5-25**	Segments are not tied to a particular server name (i.e., The segment does not hardcode a server name.)
T	**F**	**N/A**	**5-26**	The segment installation revokes the owner account's DBMS login privilege upon successful completion of database installation so that no owner accounts can be used to connect to the database.
T	**F**	**N/A**	**5-27**	Owner accounts are not used to connect to databases except during segment installation.
T	**F**	**N/A**	**5-28**	Database owner accounts do not have database administrator privileges.
T	**F**	**N/A**	**5-29**	Separate segments are provided that create required database dependencies. These segments are executed by the owning database(s).
T	**F**	**N/A**	**5-30**	The segment installation requires the owner account password to be changed upon completion.
T	**F**	**N/A**	**5-31**	Segments do not modify the core DBMS instance's configuration.
T	**F**	**N/A**	**5-32**	The segment does not assume any particular disk configuration when creating data files.
T	**F**	**N/A**	**5-33**	Any modified versions of DBMS COE tools reside with the application's client segment.
T	**F**	**N/A**	**5-34**	Scripts are provided for the DBA's use to add, modify, and remove user privileges. These scripts are documented and the documentation is submitted to the SSA with the segment.
T	**F**	**N/A**	**5-35**	The segment does not modify another segment's database schema.
T	**F**	**N/A**	**5-36**	Grants are not made to public or general-purpose users (e.g. Oracle's `PUBLIC` user).
T	**F**	**N/A**	**5-37**	Only the owner and the DBA are able to administer grants.
T	**F**	**N/A**	**5-38**	Operations that set or redirect the user's DBMS environment variables take place only within the application's execution space.
T	**F**	**N/A**	**5-39**	No indexes are created on another segment's database tables.
T	**F**	**N/A**	**5-40**	Application-level permissions are not granted to DBA accounts or to database roles used for DBMS administration.

T	**F**	**N/A**	**5-41**	Database segments are identified as unique or sharable according to their potential for sharing.
Web Services				
T	**F**	**N/A**	**5-42**	The segment's HTML files are in the segment's `$DATA_DIR/local/`*`SegDir`*`/pub` directory.
T	**F**	**N/A**	**5-43**	The segment supports HTML 3.2 and complies with style specifications (see the *DII Style Guide*) for Web applications.
T	**F**	**N/A**	**5-44**	The segment provides a notification to "disadvantaged" users if they are using a browser that does not support the features provided by the segment.
Runtime Environment				
T	**F**	**N/A**	**5-45**	The segment is launched from the same desktop provided with the COE.
T	**F**	**N/A**	**5-46**	The desktop is configured in accordance with the *DII Style Guide*.
T	**F**	**N/A**	**5-47**	The segment uses relative pathname s for files within the segment.
T	**F**	**N/A**	**5-48**	The segment does not use the "~" character, for Unix or its NT equivalent, for referencing pathnames in environment extension files which become a part of the global runtime environment.
T	**F**	**N/A**	**5-49**	The segment does not use any reserved symbols as its own from the *I&RTS* Chapter 5 unless authorized to do so by the DII COE Chief Engineer.
T	**F**	**N/A**	**5-50**	The segment does not override or alter any environment variable that it doesn't create. The segment may extend environment variables set by the affected account group through environment extension files.
T	**F**	**N/A**	**5-51**	The segment completely separates the development environment from the runtime environment, and no development environment tools, scripts, or other executables are required at runtime.
T	**F**	**N/A**	**5-52**	The segment uses the same global runtime environment configuration as provided by the COE, with extensions, if any, made through the appropriate environment extension files and segment descriptors.
T	**F**	**N/A**	**5-53**	The segment only listens on assigned ports, only registers assigned RPC addresses, and only adds assigned system UIDs (Unix).
T	**F**	**N/A**	**5-54**	The segment home environment variable points to the segment's home directory. The name of the environment variable is `segprefix_HOME` where `segprefix` is the segment's assigned prefix.
T	**F**	**N/A**	**5-55**	The segment does not add a "home" environment variable to the affected account group. It uses only "local" home environment variables, if at all.

COE-Component Segments				
T	**F**	**N/A**	**5-56**	The segment has been authorized as a COE-component segment by the DII COE Chief Engineer.
T	**F**	**N/A**	**5-57**	Segments in the kernel COE fully specify dependencies upon the components in the bootstrap COE. This is done through the `Requires` segment descriptor.

T	**F**	**N/A**	**5-58**	If a parent COE-component segment, the following environment variables (as appropriate for a Unix versus NT environment) are automatically defined as specified by this document:
				`DATA_DIR`
				`LD_LIBRARY_PATH`
				`LOGNAME`
				`LOG_NAME`
				`LOGIN_NAME`
				`MACHINE_CPU`
				`MACHINE_OS`
				`path`
				`TMPDIR`
				`TZ`
				`XAPPLRESDIR`
				`XFONTSDIR`
				`XENVIRONMENT`

T	**F**	**N/A**	**5-59**	If a COE child segment, the segment does not alter the Unix `path` environment variable. (This rule does not apply to the parent COE-component segment.)
T	**F**	**N/A**	**5-60**	All executables and other public symbols use the segment prefix unless otherwise approved by the DII COE Chief Engineer. (Certain legacy segments may be "grandfathered" by the DII COE Chief Engineer.)
Account Groups				
T	**F**	**N/A**	**5-61**	If the account group is part of the COE, prior approval has been received from the DII COE Chief Engineer to create an account group segment.
T	**F**	**N/A**	**5-62**	If the account group is a mission-application segment, prior approval has been received from the Chief Engineer to create an account group segment.
T	**F**	**N/A**	**5-63**	The environment settings from `/h/COE/Scripts` are automatically included in the runtime environment of the account group being created. (In Unix this may be accomplished by "sourcing" `/h/COE/Scripts/.cshrc.COE`.)

T	**F**	**N/A**	**5-64**	The segment provides an executable in the `Scripts` subdirectory, named `Run`*`segprefix`* where *`segprefix`* is the segment prefix, to initiate execution of the account group's applications.

T	**F**	**N/A**	**5-65**	The following environment variables , as appropriate for NT versus Unix, are defined:
				`COE_SYS_NAME`
				`DISPLAY`
				`HOME`
				`path`
				`SHELL`
				`TERM`
				`USER`
				`USER_HOME`
				`USER_DATA`
				`USER_PROFILE`

T	**F**	**N/A**	**5-66**	(Unix) The segment provides files of the form `filename.`*`segprefix`* for all environment files that segments may reference or extend through the `ReqrdScripts` descriptor.
T	**F**	**N/A**	**5-67**	(NT) The segment establishes any required global environment settings in the registry.
				Aggregate Segments
T	**F**	**N/A**	**5-68**	If a parent segment, the segment does not specify a dependency on any of its child segments.
T	**F**	**N/A**	**5-69**	If a child segment, the segment does not specify a dependency on its parent segment nor any other children in the aggregate.
T	**F**	**N/A**	**5-70**	Only one segment in the aggregate is designated as the parent.
				Segment Descriptors
T	**F**	**N/A**	**5-71**	The segment uses `SegInfo` or individual segment descriptor files, but not both.
T	**F**	**N/A**	**5-72**	The segment describes all background processes, if any, through the `Processes` descriptor.
T	**F**	**N/A**	**5-73**	All segment dependencies and conflicts are fully declared through the appropriate descriptor. (Mission-application segments need not specify dependencies on segments contained in the kernel COE unless they are version sensitive. COE-component segments need not specify dependencies on the kernel COE unless they are sensitive to version changes in the kernel COE.)
T	**F**	**N/A**	**5-74**	Memory and disk space requirements are fully and accurately specified in the `Hardware` descriptor file.

T	**F**	**N/A**	**5-75**	If not a permanent segment, the `DEINSTALL` script and `Comm.deinstall` descriptor have been fully tested to ensure they correctly make the changes indicated and completely restore the system to the state it was in prior to loading the segment.
T	**F**	**N/A**	**5-76**	The segment `Community` and `Comm.deinstall` (if applicable) descriptors have been fully tested to ensure that they correctly makes the changes indicated, and that they do not inadvertently destroy settings that may have been made by another segment. [3]
T	**F**	**N/A**	**5-77**	The `ReqrdScripts` descriptor contains no more than 32 script names and no script name is longer than 32 characters.
T	**F**	**N/A**	**5-78**	(Unix) The `PostInstall`, `PreInstall`, and `DEINSTALL` scripts have been checked and verified to *not* do a Unix `mv` across file partitions.
T	**F**	**N/A**	**5-79**	(NT) Unless a COTS segment, the segment uses the `Processes` descriptor to create boot time processes. It does not set the`Run` or `RunOnce` keys underneath `CurrentVersion`.
T	**F**	**N/A**	**5-80**	(NT) The segment's executable descriptors use the `.EXE` extension for compiled executables and `.BAT` for batch files.
T	**F**	**N/A**	**5-81**	(NT) The segment uses `SegInfo` and *not* individual segment descriptor files.
Process Compliance				
T	**F**	**N/A**	**5-82**	The segment has been registered with the SSA.
T	**F**	**N/A**	**5-83**	The DII COE Chief Engineer has granted prior approval for background, boot, RunOnce, and periodic processes.
T	**F**	**N/A**	**5-84**	System resources required by the segment have been registered with the SSA.
T	**F**	**N/A**	**5-85**	The segment prefix being used is the prefix assigned at segment registration time.
T	**F**	**N/A**	**5-86**	The ports, UIDs (Unix), and RPC addresses being used are those assigned at segment registration time.
T	**F**	**N/A**	**5-87**	The platforms and operating systems on which the segment can run have been identified and documented in a *Version Description Document*, or its equivalent.
T	**F**	**N/A**	**5-88**	All COTS products required, including the required version, are documented in the *Version Description Document* or its equivalent.
T	**F**	**N/A**	**5-89**	All required licenses are provided to the SSA with the segment, or negotiations have been made with the SSA to use licenses procured by the SSA.

[3] Developers should generally use `$APPEND` to add to community files, rather than `$DELETE` or `$REPLACE`. Developers should ensure that they delete or replace only those entries to a community file that their segment would have added.

T	**F**	**N/A**	**5-90**	Segment dependencies are noted in the *Version Description Document* or its equivalent.
T	**F**	**N/A**	**5-91**	The *Version Description Document*, or its equivalent, has been submitted with the segment to the SSA.
T	**F**	**N/A**	**5-92**	The segment has been submitted to and a ccepted for inclusion in the SSA's on line library.
T	**F**	**N/A**	**5-93**	The `VERSION` descriptor has been updated from the previous release in accordance with the requirements specified in Chapter 5. (This does not apply to the initial release of the segment.)
T	**F**	**N/A**	**5-94**	The segment is submitted with an annotated output from `VerifySeg`. All warnings are explained in full in `VSOutput`.[4]
T	**F**	**N/A**	**5-95**	The segment is submitted with a set of integration notes (`IntgNotes`) as described in Chapter 5.
T	**F**	**N/A**	**5-96**	The segment has been loaded and tested in the COE environment prior to submission to the SSA.
T	**F**	**N/A**	**5-97**	Segment installation has been tested through the same installation tools used by site operators. (`TestInstall` alone does *not* satisfy this requirement. The `COEInstaller` tool must be used to load and remove the segment.)
T	**F**	**N/A**	**5-98**	If removable, the segment has been tested and confirmed that it can be successfully removed from the system.
T	**F**	**N/A**	**5-99**	If special installation/integration procedures/problems exist, then they are incorporated into the `PostInstall` (or other) descriptors as appropriate, and documented in the `IntgNotes` descriptor file.
T	**F**	**N/A**	**5-100**	(NT) Unless a COTS segment or authorized by the DII COE Chief Engineer , the segment does not register "uninstall" information in the registry (e.g., subkey `CurrentVersion\Uninstall`). ("uninstall" information is handled automatically for the segment by the COE installation tools.)
T	**F**	**N/A**	**5-101**	(NT) If an approved segment registers "uninstall" information in the registry, the `$USES_UNINSTALL` keyword is declared in the segment's `Direct` descriptor.
Miscellaneous				
T	**F**	**N/A**	**5-102**	The segment creates and initializes dynamic data files that are updated as the system executes (e.g., message logs, operator preferences). If an expected file is missing, the segment generates a runtime error message and gracefully terminates with an appropriate message to the operator.
T	**F**	**N/A**	**5-103**	If a patch segment, it follows the patch segment naming convention.

[4] This can be done by redirecting the output of `VerifySeg` to the file `VSOutput`. Then, use any convenient ASCII editor to edit `VSOutput` to insert comments to explain all warning messages.

T	**F**	**N/A**	**5-104**	The segment does not alter *any* files outside its own directory with the following exceptions: (a) the segment is a patch segment and must modify files in another segment; (b) the segment is creating temporary files or directories in directories established for temporary storage; (c) the segment is modifying files created for it by the operating system; or (d) the files are created or modified through approved APIs or segment descriptors during segment installation.
T	**F**	**N/A**	**5-105**	The segment does not create copies of executables from other segments. (There are rare instances where this may be required to create a patch segment. Such exceptions require the prior approval of the Chief Engineer.)
T	**F**	**N/A**	**5-106**	The segment does not contain any circular dependencies (e.g., Seg A depends on Seg B, Seg B depends on Seg C, Seg C depends on Seg A is not allowed).
T	**F**	**N/A**	**5-107**	The segment does not delete itself via the `DEINSTALL` descriptor, nor perform any other operations that are handle d by the COE installation tools (e.g., undo changes made to community files).

B-6. Intermediate DII Compliance (Level 6)

Security				
T	**F**	**N/A**	**6-1**	The segment satisfies at least one of the following two requirements: (1) The segment contains only subdirectories directly underneath the segment 's home directory. All files are at least one level down from the segment's home directory. (2) The segment has no directories or files that have the equivalent of the Unix 777 file permissions.
T	**F**	**N/A**	**6-2**	If the data for a particular segment contains any classified entries, then all of its data is packaged in a separate data segment and classified accordingly.
T	**F**	**N/A**	**6-3**	Classified applications segments are packaged separately from unclassified segments, or from segments which are classified at a lower level. (It is permissible to create aggregate segments that contain segments at different classification levels, but the aggregate must be labeled with the highest classification level of any segment within the aggregate.)
T	**F**	**N/A**	**6-4**	Termination of segment execution, whether premature, inadvertent, or intentional does not place the operator at a command-line prompt.
T	**F**	**N/A**	**6-5**	The Chief Engineer has authorized privileged processes in the segment. The `$PRIVPROC` keyword is stated in the `Direct` segment descriptor, and the privileged processes are listed in the `Processes` segment descriptor.
T	**F**	**N/A**	**6-6**	(Unix) The segment does not contain any shell scripts that `SUID` or `SGID` to root.
Standards Compliance				
T	**F**	**N/A**	**6-7**	The segment is either completely comp liant with the *DII Style Guide* or has minimal deviations that have been approved by the Chief Engineer .
T	**F**	**N/A**	**6-8**	The segment is available on all COE-supported platforms unless otherwise approved by the Chief Engineer .
T	**F**	**N/A**	**6-9**	The segment does not alter any community files except through COE segment descriptors or published APIs.
T	**F**	**N/A**	**6-10**	The segment does not use directories with different names than specified in Chapter 5 to fulfill the purpose of `Scripts`, `bin`, `data`, etc. (`progs` and `libs` are acceptable for this level for as long as the COE tools support them.)
T	**F**	**N/A**	**6-11**	If the segment contains APIs written in C, the header files for the public APIs are ANSI-C-compliant and use function prototypes, and the header files are constructed to support C++ calling routines as described in Chapter 9.
T	**F**	**N/A**	**6-12**	Specification files for Ada are included for all APIs, unless the Chief Engineer has granted a waiver.

T	**F**	**N/A**	**6-13**	(NT) All `INI` files used that are local to the segment are stored in the segment's `data/INI` subdirectory.
T	**F**	**N/A**	**6-14**	(NT) The segment supports long filenames.
T	**F**	**N/A**	**6-15**	(NT) The segment uses filename extensions in accordance with standard Windows usage (`TXT` for ASCII files, `DLL` for dynamic link libraries, etc.).
GUI Environment				
T	**F**	**N/A**	**6-16**	(Unix) The segment does not alter any X or Motif supplied files (e.g., `Xdefaults`, `rgb.txt`).
Database Services				
T	**F**	**N/A**	**6-17**	Oracle Public Synonyms are not used.
T	**F**	**N/A**	**6-18**	Database segments do not create user accounts, except for a database services account.
T	**F**	**N/A**	**6-19**	Grants are made to database roles/groups, not user accounts or general-purpose users (e.g., Oracle's `PUBLIC` user).
T	**F**	**N/A**	**6-20**	The application does not assume the existence of any particular user.
T	**F**	**N/A**	**6-21**	Data elements do not use machine-dependent data types.
T	**F**	**N/A**	**6-22**	The segment does not create data objects in other segments except through documented inter-database dependencies (e.g., triggers) and published APIs.
T	**F**	**N/A**	**6-23**	External object dependencies are listed under the `Database` descriptor.
T	**F**	**N/A**	**6-24**	The segment uses only the DBMS provided by the COE, or has an approved migration plan.
T	**F**	**N/A**	**6-25**	The segment either implements DOD 8320 data standards, or has an approved plan for doing so. (The migration plan must be coordinated with the DII COE Chief Engineer for any data fields that are part of Universal or Shared data segments. Data fields that are part of a Unique data segment do not require DII COE Chief Engineer approval.)
T	**F**	**N/A**	**6-26**	Database roles that span multiple database segments are defined in their own segments.
T	**F**	**N/A**	**6-27**	Data objects and elements follow naming conventions specified in Chapter 4.
T	**F**	**N/A**	**6-28**	Definitions for schema components are provided in the DBMS data dictionary.

Web Services				
T	**F**	**N/A**	**6-29**	The segment uses the Web server provided by the COE rather than bringing along its own Web server.
COTS Products				
T	**F**	**N/A**	**6-30**	All COTS products are packaged as separate, individual COTS segments.
T	**F**	**N/A**	**6-31**	The `PostInstall` script ensures that there is enough space in the directories where the COTS product will be installed and uses `COEInstError` to report an error message if not.
T	**F**	**N/A**	**6-32**	The `FilesList` descriptor has been validated as correctly documenting what files and directories constitute the COTS product. This does not apply to COTS products in the bootstrap COE.
Runtime Environment				
T	**F**	**N/A**	**6-33**	If the segment creates temporary files, they are deleted when no longer needed.
T	**F**	**N/A**	**6-34**	If the segment uses absolute pathnames to reference files outside the segment, it is able to determine the absolute path at runtime, and is able to handle symbolic links that are themselves symbolic links.
T	**F**	**N/A**	**6-35**	The segment reuses environment variables already defined by the COE or by the affected account group. It does not create any environment variables that are identical in value to those defined by the COE or the affected account group, or that can be derived from them.
T	**F**	**N/A**	**6-36**	The segment does not create any environment variables or other public symbols with the same name as any environment variables listed as reserved in the *I&RTS*.
T	**F**	**N/A**	**6-37**	Shared libraries (Unix) and DLLs (NT) provided by the segment are in the segment's `bin` subdirectory. The `SharedFile` descriptor is used to define them, and they are named using the segment prefix convention.
T	**F**	**N/A**	**6-38**	The segment does not insert the current working directory (e.g., ".") into the search path for executables.
T	**F**	**N/A**	**6-39**	(Unix) The global environment is extended through runtime extension files in the segment 's `Scripts` subdirectory.
T	**F**	**N/A**	**6-40**	(Unix) Fonts and app-defaults located underneath the segment 's `data` subdirectory follow the segment prefix naming convention specified in the *I&RTS*.
T	**F**	**N/A**	**6-41**	(Unix) The segment appends, not prepends, its `bin` subdirectory to the environment variable used for the search path for finding executables. (This does not apply to COE child segments.)

T	**F**	**N/A**	**6-42**	(Unix) The segment uses relative pathnames for symbolic links used to reference files within the segment.
T	**F**	**N/A**	**6-43**	(NT) The segment uses relative pathnames for shortcuts used to reference files within the segment.
T	**F**	**N/A**	**6-44**	(NT) The segment stores its DLL files in the segment's `bin` subdirectory.
T	**F**	**N/A**	**6-45**	(NT) Unless a COTS segment, the segment does not alter the Windows `path` environment variable.
T	**F**	**N/A**	**6-46**	(NT) The segment does not use polling as a synchronization technique.
T	**F**	**N/A**	**6-47**	(NT) The segment does not use MS-DOS functions.
Segment Descriptors				
T	**F**	**N/A**	**6-48**	The `ReleaseNotes` descriptor conforms to the requirements stipulated in Chapter 5.
T	**F**	**N/A**	**6-49**	If any files need special permission/ownership settings, they are established through the `FileAttribs` descriptor if the descriptor supports the required setting. Exceptions to this are documented and approved by the Chief Engineer.
Process Compliance				
T	**F**	**N/A**	**6-50**	The segment includes an API test suite that exhaustively exercises all APIs provided by the segment.
T	**F**	**N/A**	**6-51**	The segment includes man pages , or HTML-format pages, for all APIs that are to be distributed with the Developer's Toolkit.
T	**F**	**N/A**	**6-52**	The segment has been compiled without the debug option enabled .
T	**F**	**N/A**	**6-53**	If the segment has published APIs implemented as shared libraries, static libraries are provided as well.
T	**F**	**N/A**	**6-54**	If the segment uses another segment's public APIs and they are implemented as shared libraries, the segment is submitted linked with the shared libraries and not the static libraries.
T	**F**	**N/A**	**6-55**	If the segment has a `DEINSTALL` and `Community` descriptor, it also includes a `Comm.deinstall` descriptor which reverses the actions of the `Community` descriptor during segment removal.
T	**F**	**N/A**	**6-56**	The segment has been tested to ensure that it successfully installs over and replaces any previous version of the segment.
T	**F**	**N/A**	**6-57**	If the segment contains a large static database, it is provided as a separate data segment.
T	**F**	**N/A**	**6-58**	(Unix) The segment executables have been run through the Unix `strip` program.

Miscellaneous				
T	**F**	**N/A**	**6-59**	Unless an account group segment, the segment is integrated within one or more of the predefined account groups.
T	**F**	**N/A**	**6-60**	If the COE provides functions required by the segment, at least 50% of the functions required are provided by the COE and not by duplicative code in the segment.
T	**F**	**N/A**	**6-61**	API backwards compatibility conforms to the version numbering scheme described in Chapter 3.
T	**F**	**N/A**	**6-62**	The segment does not provide access to a command-line prompt, except with prior Chief Engineer approval.

B-7. Interoperable Compliance (Level 7)

Security				
T	**F**	**N/A**	**7-1**	The segment does not place any temporary files in the system maintained temporary directory that are sensitive to alteration, deletion, or disclosure to unauthorized users.
T	**F**	**N/A**	**7-2**	If the segment creates files that are sensitive to alteration or deletion by unauthorized users, they are not placed in any directory where such users have write access, and those files do not have write permissions set for such users.
T	**F**	**N/A**	**7-3**	If the segment creates files which are sensitive to disclosure to unauthorized users, they are not placed in any directory where users have such access.
T	**F**	**N/A**	**7-4**	Entering a command-line mode requires the operator to enter a password and forces execution of the system login process.
T	**F**	**N/A**	**7-5**	The segment does not contain features with multiple security levels, unless an aggregate segment.
T	**F**	**N/A**	**7-6**	Unclassified sample data is provided with the segment to allow for unclassified testing and training.
T	**F**	**N/A**	**7-7**	The segment does not create files or directories with write permissions for "world" users, except as authorized by the Chief Engineer.
T	**F**	**N/A**	**7-8**	Data files with different file permissions are split into separate directories underneath the segment's `data` subdirectory.
Standards Compliance				
T	**F**	**N/A**	**7-9**	If written in C, the segment is ANSI-C-compliant.
T	**F**	**N/A**	**7-10**	If written in Ada, the segment is Ada-95-compliant unless otherwise authorized by the Chief Engineer.
T	**F**	**N/A**	**7-11**	If the segment contains public APIs, Ada and C interfaces are both provided unless the Chief Engineer grants a waiver.
T	**F**	**N/A**	**7-12**	Global and local data owned by the segment are located underneath `$DATA_DIR` as described in Chapter 5.
T	**F**	**N/A**	**7-13**	Operator-specific data is located underneath `/h/USERS` as described in Chapter 5.
T	**F**	**N/A**	**7-14**	Excepting COTS segments, all environment variables are named with the segment prefix unless approved by the Chief Engineer. (The Chief Engineer may authorize "grandfathering" of certain environment variables.)
T	**F**	**N/A**	**7-15**	The segment uses only POSIX.1-defined interfaces to access the operating system, unless authorized by the Chief Engineer.

Network Services				
T	**F**	**N/A**	**7-16**	(NT) The segment determines the location for shared data through the registry.
T	**F**	**N/A**	**7-17**	(NT) The segment stores information about shared resources in the location specified in Chapter 6.
GUI Environment				
T	**F**	**N/A**	**7-18**	The segment uses resource files to control window behavior rather than hardcoded window behavior attributes.
T	**F**	**N/A**	**7-19**	The segment supports cut and paste between GUI-based segments through the use of a shared clipboard.
T	**F**	**N/A**	**7-20**	(NT) The segment uses TrueType fonts.
Database Services				
T	**F**	**N/A**	**7-21**	Data objects within the segment do not duplicate those already contained in available Universal database segments.
T	**F**	**N/A**	**7-22**	Database fragmentation schemas are contained in separate segments.
T	**F**	**N/A**	**7-23**	Database roles/groups are specific to application privileges, not general purpose.
T	**F**	**N/A**	**7-24**	The segment does not duplicate any data available from the SHADE repository, except for performance reasons, unless approved by the DII COE Chief Engineer.
T	**F**	**N/A**	**7-25**	The segment uses only FIPS-127-2 SQL-defined interfaces to access the RDBMS query services.
T	**F**	**N/A**	**7-26**	Data object creation script files follow the specified structure and naming convention.
T	**F**	**N/A**	**7-27**	The data objects contained within a database segment are standardized according to DOD 8320 guidance.
T	**F**	**N/A**	**7-28**	All constraints and business rules are in the database, not the applications.
T	**F**	**N/A**	**7-29**	The database server segment provides a reload capability and a non-destructive update capability.

DCE Services				
T	**F**	**N/A**	**7-30**	If the application uses DCE services, only the DCE interfaces defined by the DCE version supported by the COE (see Appendix A) are used to access those services.
Runtime Environment				
T	**F**	**N/A**	**7-31**	The segment does not include any environment variables that could be derived from an already defined environment variable.
T	**F**	**N/A**	**7-32**	Segment references to global and local data are done through the `$DATA_DIR` environment variable.
T	**F**	**N/A**	**7-33**	(NT) The segment stores private INI files, if any, in the segment's `data\INI` subdirectory.
Miscellaneous				
T	**F**	**N/A**	**7-34**	The segment does not duplicate any functions provided by COE-component segments unless approved by the DII COE Chief Engineer.
T	**F**	**N/A**	**7-35**	No more than 25% of the segment's accesses to COE-component segments is through private APIs.
T	**F**	**N/A**	**7-36**	(NT) The segment does not duplicate any Windows functions.

B-8. Full DII Compliance (Level 8)

Security				
T	**F**	**N/A**	**8-1**	Entry to and exit from the command-line mode causes an entry into the system audit logs that specifies the date, time, and user involved.
T	**F**	**N/A**	**8-2**	Information written to the audit log includes the segment prefix.
T	**F**	**N/A**	**8-3**	The segment does not mix restricted and unrestricted data files in the same directory.
Standards Compliance				
T	**F**	**N/A**	**8-4**	The segment does not use any conventions obsoleted by this document (use of `progs` vs. `bin`, use of `COMPONENT` vs. `CHILD`, use of `ModName` and `SegType` vs. `SegName` etc.).
T	**F**	**N/A**	**8-5**	All public symbols are named with the segment prefix naming convention.
T	**F**	**N/A**	**8-6**	All directory and file names begin with an alphanumeric character.
T	**F**	**N/A**	**8-7**	The segment follows the convention that data owned by the segment under `$DATA_DIR` is in the form `$DATA_DIR/local/`*`segdir`*`/data` and `$DATA_DIR/global/`*`segdir`*`/data` where *`segdir`* is the segment's home directory name.
GUI Environment				
T	**F**	**N/A**	**8-8**	The segment is fully compliant with the *DII Style Guide*.
T	**F**	**N/A**	**8-9**	(NT) The segment uses common control and dialog functions from `COMCTL32.DLL` and `COMDLG32.DLL`.
T	**F**	**N/A**	**8-10**	(NT) The segment is close aware.
T	**F**	**N/A**	**8-11**	(NT) The segment uses the Windows print dialog box for selecting printer configuration parameters.

Database Services				
T	**F**	**N/A**	**8-12**	Data elements are chosen from Joint standards and use the data type and units of measure prescribed in the standard.
T	**F**	**N/A**	**8-13**	A test database is provided together with test procedures to verify correct installation of the database and associated roles, and to verify correct operation of constraints defined in the database.
T	**F**	**N/A**	**8-14**	The segment does not duplicate any data already maintained in the SHADE repository *or* the COE-based target system, unless for performance reasons and only as approved by the DII COE Chief Engineer.
T	**F**	**N/A**	**8-15**	The segment uses only the DBMS provided by the COE.
Runtime Environment				
T	**F**	**N/A**	**8-16**	The segment adds no more than one "home" environment variable to the global environment.
T	**F**	**N/A**	**8-17**	All executables and public symbols are named *segprefix*_name, where *segprefix* is the assigned segment prefix.
T	**F**	**N/A**	**8-18**	(NT) Local environmental settings are established through an LOCALENV.BAT file in the segment's Scripts subdirectory.
Segment Descriptors				
T	**F**	**N/A**	**8-19**	The segment uses SegInfo rather than individual segment descriptor files.
Process Compliance				
T	**F**	**N/A**	**8-20**	The segment includes a set of test data for verifying correct segment operation.
Miscellaneous				
T	**F**	**N/A**	**8-21**	The segment does not use any private APIs to access external segments. All accesses are through public APIs or approved protocol standards.
T	**F**	**N/A**	**8-22**	Operator data is located through the *Preferences* APIs.
T	**F**	**N/A**	**8-23**	The current operator profile is obtained through the *Preferences* APIs.
T	**F**	**N/A**	**8-24**	The segment does not duplicate functionality provided by any other segment unless approved by the DII COE Chief Engineer.

B-9. Recommended Guidelines

The items contained in the following checklist are not mandatory, but are general guidelines for most segments. They are not considered in establishing the DII compliance level.

T	**F**	**N/A**	**M-1**	The segment does not use symbolic links.
T	**F**	**N/A**	**M-2**	The segment does not use boot or background processes. It uses session-or-transient-level processes instead.
T	**F**	**N/A**	**M-3**	The segment allows comments in ASCII data files. The # character is the standard for single line comments while C style comments[5] (delimited by the /* */ pair) are the standard for all other comments.
T	**F**	**N/A**	**M-4**	If written in C, the segment has been run through `lint` to detect potential coding errors.
T	**F**	**N/A**	**M-5**	If written in C, the segment has been compiled with the `STRICT` constant.
T	**F**	**N/A**	**M-6**	The segment uses `putenv` or an equivalent technique to create segment-specific environment variables that are inherited locally, rather than adding environment variables to the global environment.
T	**F**	**N/A**	**M-7**	(Unix) The segment links with X and Motif shared libraries.

[5] C style comments for ASCII data files are recommended over Ada-style comments, not because of a preference for C over Ada, but because it allows an entire block of lines to be easily commented out. Ada-style comments would require editing each individual line to insert the "--" comment delimiter.

T	**F**	**N/A**	**M-8**	(NT) The segment uses Microsoft DLLs for required functions that are provided by NT.
T	**F**	**N/A**	**M-9**	(NT) The segment links to DLL functions by using symbolic names, not ordinal numbers.
T	**F**	**N/A**	**M-10**	(NT) The segment exports DLL functions by symbolic name, not ordinal numbers.
T	**F**	**N/A**	**M-11**	(NT) The segment does not include an `app-defaults` or `fonts` subdirectory.
T	**F**	**N/A**	**M-12**	(NT) The segment uses the Win32 API GDI for creating 2D graphics.
T	**F**	**N/A**	**M-13**	(NT) The segment uses OpenGL APIs for 3D graphics.
T	**F**	**N/A**	**M-14**	(NT) `PORTTOOL.EXE` has been used to identify potential problems with how the segment uses Windows APIs.
T	**F**	**N/A**	**M-15**	(NT) The segment operates correctly under both Windows NT and Windows 95.
T	**F**	**N/A**	**M-16**	(NT) The segment uses message crackers from `WINDOWSX.H`.

This page is intentionally blank.